Lezione 2

Fondamenti di Programmazione in Java

© 2000 McGraw-Hill

Lezione 2 - Obiettivi

Al termine di questa lezione saremo in grado di

- Identificare i componenti fondamentali di un programma in Java.
- Scrivere semplici applicazioni in Java.
- Descrivere la differenza tra dichiarazione di oggetti e creazione di oggetti.
- Descrivere il procedimento di creazione ed esecuzione di programmi in Java.
- Utilizzare le classi MainWindow e MessageBox del package javabook all'interno dei programmi.

© 2000 McGraw-Hill

Il mio Primo Programma in Java

- Un programma per mostrare una finestra sullo schermo.
- La dimensione della finestra è leggermente inferiore a quella dello schermo, e la finestra è posizionata al centro dello schermo con il titolo Sample Java Application.
- Il programma illustra il concetto fondamentale della programmazione ad oggetti:

Un programma scritto in un linguaggio ad oggetti usa degli oggetti.

© 2000 McGraw-Hill

Programma MyFirstApplication

```
/*
   Programma MyFirstApplication

   Questo programma mostra una finestra sullo schermo. La
   finestra è al centro dello schermo, e la dimensione della
   finestra è leggermente inferiore a quella dello schermo.

*/

import javabook.*;

class MyFirstApplication
{
   public static void main(String[ ] args)
   {
      MainWindow     mainWindow;
      mainWindow = new MainWindow();
      mainWindow.setVisible( true );
   }
}
```

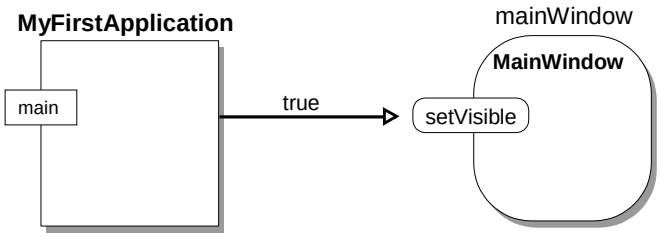
Dichiarazione

Creazione

Messaggio

© 2000 McGraw-Hill

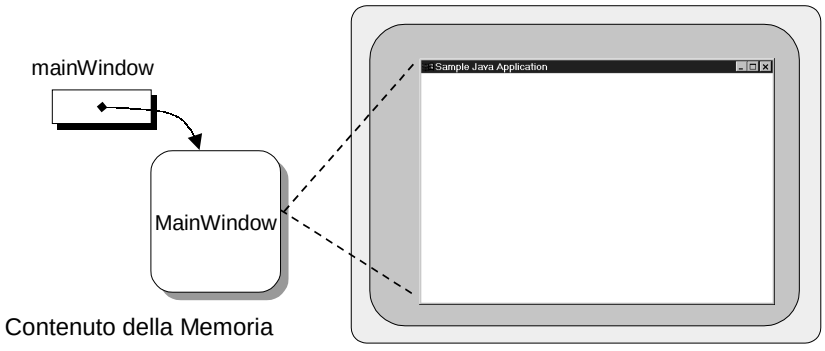
Diagramma di MyFirstApplication



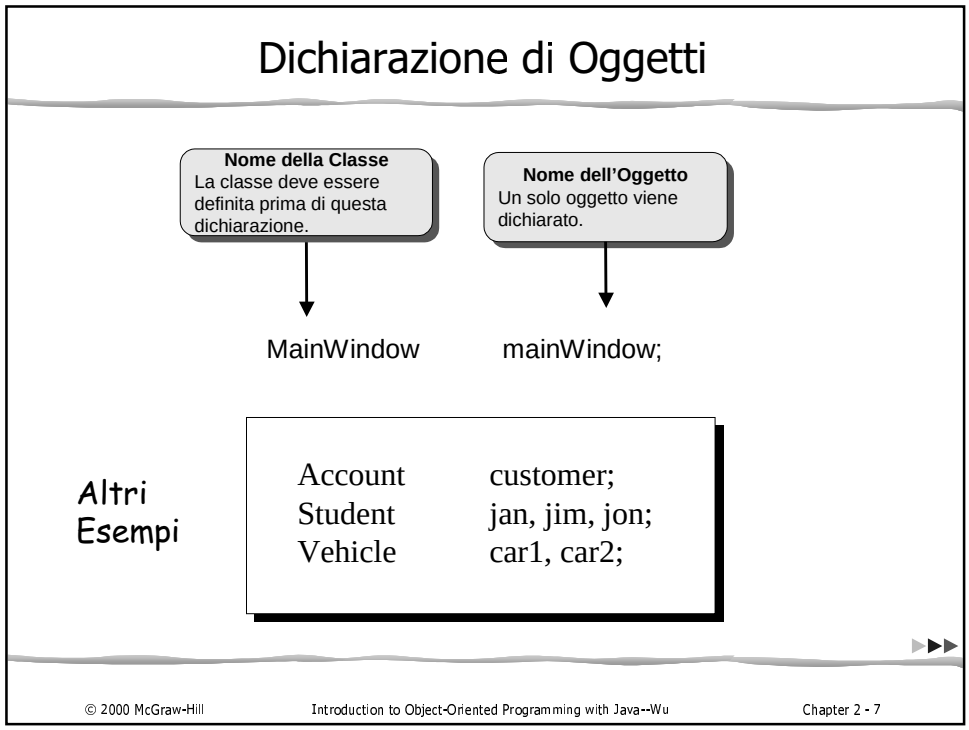
© 2000 McGraw-Hill Introduction to Object-Oriented Programming with Java--Wu Chapter 2 - 5

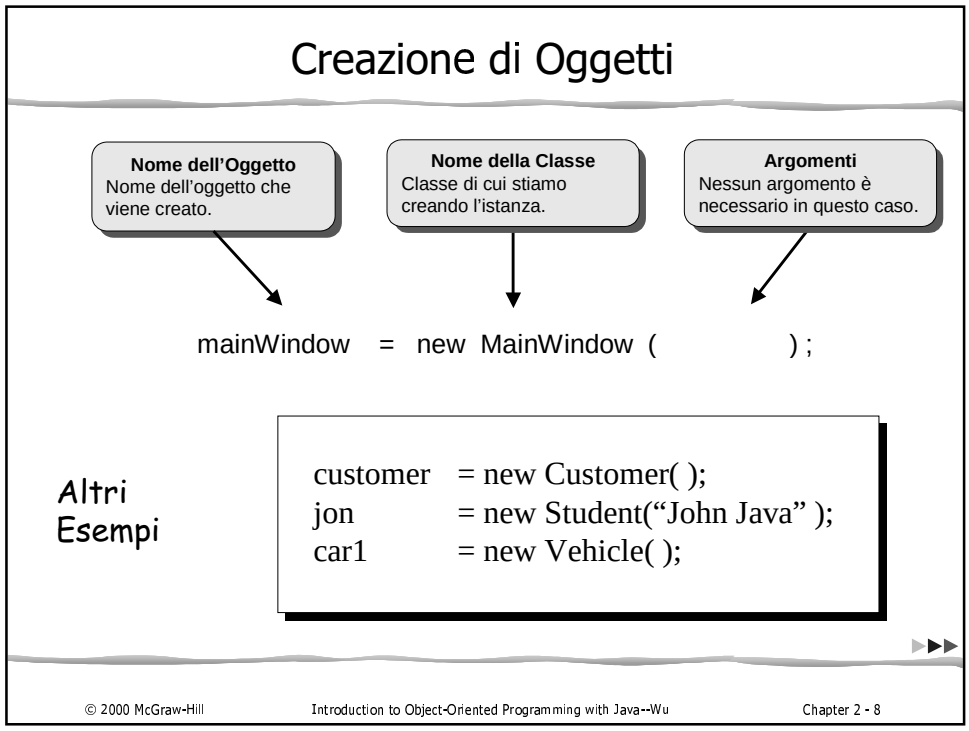
Flusso di MyFirstApplication

```
MainWindow  mainWindow;
mainWindow = new MainWindow();
mainWindow.setVisible( true );
```



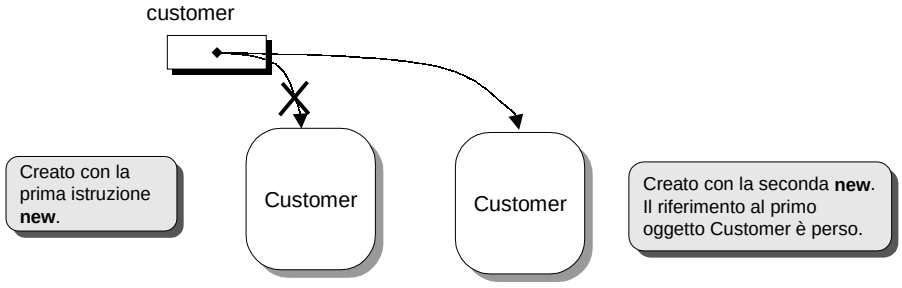
© 2000 McGraw-Hill Introduction to Object-Oriented Programming with Java--Wu Chapter 2 - 6


Dichiarazione di Oggetti
Nome della Classe
La classe deve essere definita prima di questa dichiarazione.
Nome dell'Oggetto
Un solo oggetto viene dichiarato.
MainWindow
mainWindow;
Altri Esempi
Account customer;
Student jan, jim, jon;
Vehicle car1, car2;
© 2000 McGraw-Hill
Introduction to Object-Oriented Programming with Java--Wu
Chapter 2 - 7


Creazione di Oggetti
Nome dell'Oggetto
Nome dell'oggetto che viene creato.
Nome della Classe
Classe di cui stiamo creando l'istanza.
Argomenti
Nessun argomento è necessario in questo caso.
mainWindow = new MainWindow ( ) ;
Altri Esempi
customer = new Customer( );
jon = new Student("John Java");
car1 = new Vehicle( );
© 2000 McGraw-Hill
Introduction to Object-Oriented Programming with Java--Wu
Chapter 2 - 8

Distinzione tra Dichiarazione e Creazione

```
Customer   customer;
customer   =  new  Customer( );
customer   =  new  Customer( );
```



Invio di Messaggi

Nome dell'Oggetto
Nome dell'oggetto a cui il messaggio è inviato.

Nome del Metodo
Nome del messaggio che stiamo inviando.

Argomento
Argomento passato con il messaggio.

```
mainWindow . setVisible ( true ) ;
```

Altri Esempi

```
account.deposit( 200.0 );
student.setName("john");
car1.startEngine( );
```

Componenti di un Programma

- Un programma Java è composto da
  - commenti,
  - direttive import, e
  - definizioni di classi.

© 2000 McGraw-Hill

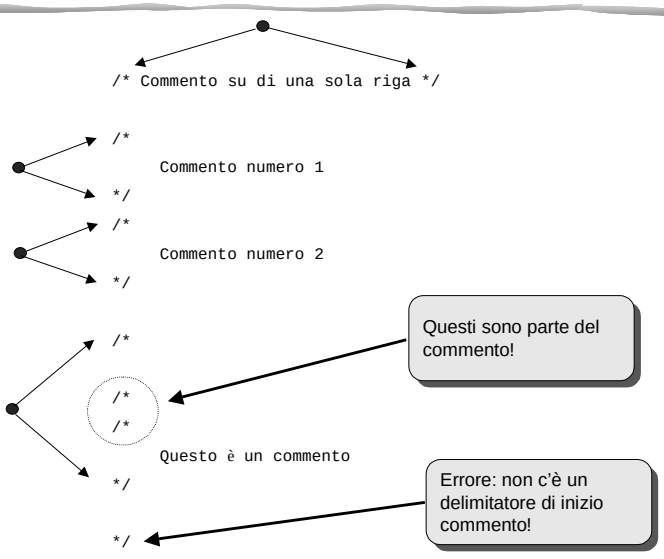
Componenti: Commenti

```
/*
    Programma MyFirstApplication

    Questo programma mostra una finestra sullo schermo. La
    finestra è al centro dello schermo, e la dimensione della
    finestra è leggermente inferiore a quella dello schermo.

*/

import javabook.*;

class MyFirstApplication
{
   public static void main(String[ ] args)
   {
      MainWindow     mainWindow;
      mainWindow = new MainWindow();
      mainWindow.setVisible( true );
   }
}
```

Commento

© 2000 McGraw-Hill

Delimitazione dei Commenti



Tre Tipi di Commenti

```
/*
     Un commento disposto
     su tre linee di
     testo.
*/
```

Commento su più linee

```
// Un commento
// Un altro commento
// Un terzo commento
```

Commenti su linea singola

```
/**
 * Questa classe fornisce le funzioni basilari di orologio.
 * Oltre all'ora corrente e alla data odierna si può
 * usare questa classe come cronometro.
 */
```

Commenti javadoc

Componenti: Direttive Import

```
/*
   Programma MyFirstApplication

   Questo programma mostra una finestra sullo schermo. La
   finestra è al centro dello schermo, e la dimensione della
   finestra è leggermente inferiore a quella dello schermo.

*/
import javabook.*;
class MyFirstApplication
{
   public static void main(String[ ] args)
   {
      MainWindow     mainWindow;
      mainWindow = new MainWindow();
      mainWindow.setVisible( true );
   }
}
```

Direttiva Import

© 2000 McGraw-Hill

Sintassi e Semantica di Import

Nome del Package
Nome del package che definisce le classi necessarie.

Nome della Classe
Nome della classe da importare. L'asterisco corrisponde a tutte le classi.

<package name> . <class name> ;

e.g. javabook . InputBox;

Altri Esempi

```
import    javabook.*;
import    java.awt.image.ColorModel;
import    com.drcaffeine.galapagos.*;
```

© 2000 McGraw-Hill

Componenti: Definizioni delle Classi

```
/*
   Programma MyFirstApplication

   Questo programma mostra una finestra sullo schermo. La
   finestra è al centro dello schermo, e la dimensione della
   finestra è leggermente inferiore a quella dello schermo.

*/

import javabook.*;

class MyFirstApplication
{
   public static void main(String[ ] args)
   {
      MainWindow     mainWindow;
      mainWindow = new MainWindow();
      mainWindow.setVisible( true );
   }
}
```

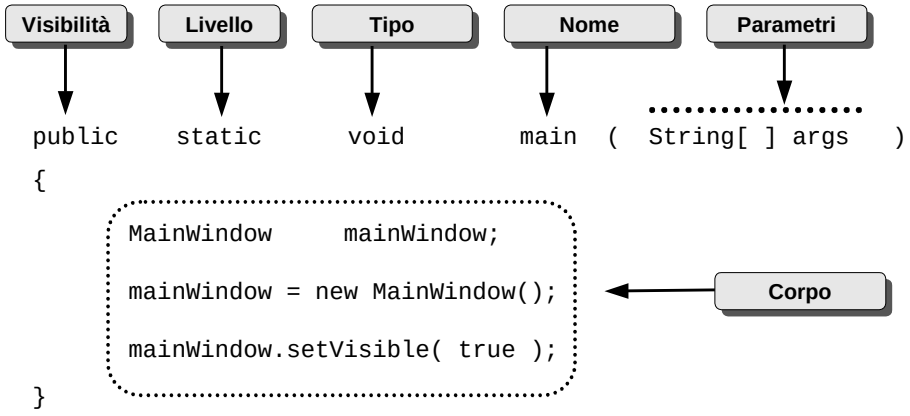
Definizione di una Classe

Componenti: Definizioni dei Metodi

```
/*
   Programma MyFirstApplication

   Questo programma mostra una finestra sullo schermo. La
   finestra è al centro dello schermo, e la dimensione della
   finestra è leggermente inferiore a quella dello schermo.

*/

import javabook.*;

class MyFirstApplication
{
   public static void main(String[ ] args)
   {
      MainWindow     mainWindow;
      mainWindow = new MainWindow();
      mainWindow.setVisible( true );
   }
}
```

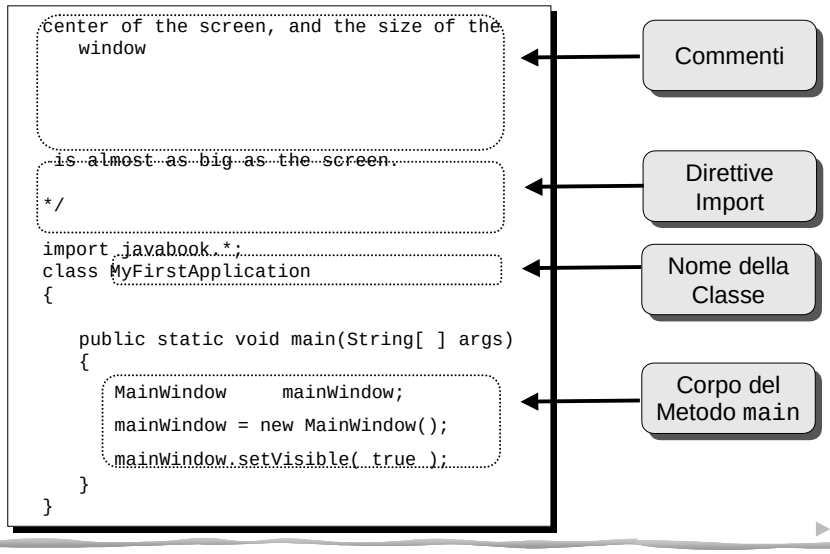
Definizione di un Metodo

Elementi della Definizione di un Metodo



© 2000 McGraw-Hill

Schema per Semplici Programmi Java



© 2000 McGraw-Hill

Programmare in Java

- Passo 1 Scrittura
  - Scrivere il programma utilizzando un editor e registrare il programma su un file detto *sorgente*.
- Passo 2 Compilazione
  - Il compilatore Java legge il file sorgente e produce un file in formato intermedio detto *bytecode*.
- Passo 3 Esecuzione
  - Il file in bytecode viene eseguito tramite una Java Virtual Machine (JVM).

© 2000 McGraw-Hill

Il Package javabook

- Il passo fondamentale nella programmazione ad oggetti è l'utilizzo di classi predefinite.
- Nei programmi di esempio sono state utilizzate le classi predefinite nel package javabook.
- Vantaggi nell'uso di javabook:
  - Mostra come i programmi vengono sviluppati in ambienti professionali.
  - Minimizza l'impatto con la sintassi e la semantica di Java.
  - Consente di scrivere programmi utilizzabili senza conoscere troppi dettagli di Java.
  - Serve come esempio per la progettazione di classi.

© 2000 McGraw-Hill

Esempio: Visualizzare un Testo

- Problema

  Scrivere un programma che visualizzi il testo

  I Love Java.

- Progetto
  - **Alternativa 1**: Impostare il titolo di un oggetto MainWindow al testo designato.
  - **Alternativa 2**: Utilizzare un oggetto MessageBox. Questo oggetto serve per visualizzare un breve testo su di una singola linea per catturare l'attenzione dell'utente. La classe MessageBox è disponibile all'interno del package javabook.

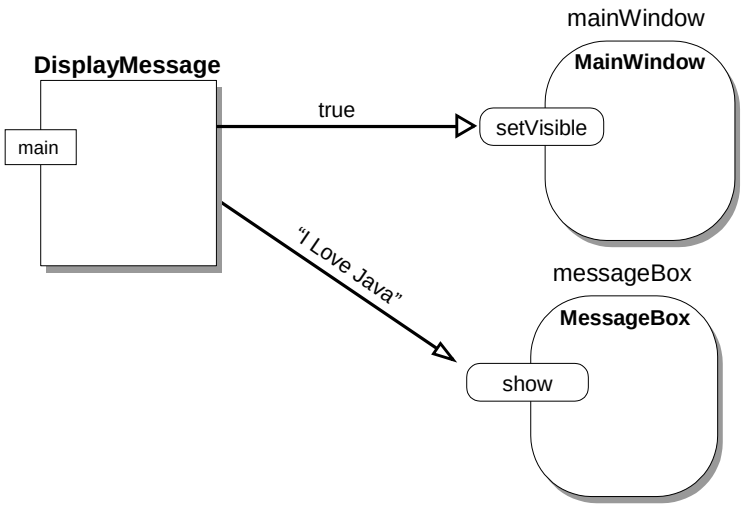
© 2000 McGraw-Hill

Esempio: Documento di Progetto

Progetto del Programma: DisplayMessage	
Classe	Scopo
DisplayMessage	La classe principale del programma.
MainWindow	La cornice principale del programma. Il titolo è impostato come "Display Message". La classe è disponibile in javabook.
MessageBox	Il dialogo per visualizzare il messaggio richiesto. La classe è disponibile in javabook.

© 2000 McGraw-Hill

Esempio: Diagramma di DisplayMessage



© 2000 McGraw-Hill

Esempio: Codice Sorgente

```
/*
   Program DisplayMessage

   Il programma mostra il testo "I Love Java". Il programma usa un
   oggetto MessageBox del package javabook per visualizzare il testo.
*/

import javabook.*;

class DisplayMessage
{
   public static void main(String[] args)
   {
       MainWindow mainWindow;           //dichiara due oggetti
       MessageBox messageBox;

       mainWindow = new MainWindow("Display Message");  //crea due oggetti
       messageBox = new MessageBox(mainWindow);

       mainWindow.setVisible( true );   //visualizza due oggetti: prima la
       messageBox.show("I Love Java");  //cornice e poi il dialogo
   }
}
```

© 2000 McGraw-Hill

Esempio: Verifica del Funzionamento

- Eseguendo il programma...